We claim:

1. A method of tracing data traffic on a network, the method comprising: at the transport layer of a protocol stack residing on a first device in the network, detecting a transmission or receipt of data to or from a second device on the network; and in response to the transmission or receipt being detected, recording the transmission or receipt as an entry in a trace log, wherein the trace log is accessible to determine the volume of data traveling over a network.

2. The method of claim 1, wherein the protocol stack is a TCP/IP stack.

3. The method of claim 1, wherein the detection step further comprises the step of detecting the presence of an input/output packet representing the transmission or receipt.

4. A method of tracing a transmission of data over a computer network comprising: detecting the presence of an input/output packet requesting a transmission; searching the input/output request packet to determine the identity of the process that created the input/output request packet; and storing in a trace log an entry representing the transmission, wherein the entry comprises the identity of the process, and wherein the trace log is accessible to determine the volume of data being transmitted over the network.

5. The method of claim 4, further comprising: detecting an acknowledgment of the transmission; and in response to the detection of the acknowledgment, storing in the trace log an entry representing the completion of the transmission.

6. A method of tracing a receipt of data from a computer network comprising: detecting the presence of a packet for an input/output connection to a port; searching the packet to determine the identity of the process that created the packet; and in response to the detection of a receipt of data at the port, storing in a trace log an entry

Cont Sub B4

representing the receipt of the data, wherein the entry comprises the process identification, and wherein the trace log is accessible to determine the volume of the data being transmitted over the network.

7. The method of claim 6, further comprising: creating a connection object representing the opening of the port connection by the process; copying the process identification from the connection object into a transport control block associated with the port; and in response to the detection of the receipt of data at the port, copying the process identification into the trace log.

8. The method of claim 7, further comprising: copying the process identification from the connection object into the transport control block so that the process identification is contiguous with the rest of the data in the transport control block.

9. The method of claim 8, further comprising: detecting the presence of an input/output request packet indicating that the data receipt is complete; and in response to the detection of the completion input/ouput request packet, making an entry representing the receipt of the data into a trace log.

Sub B5

10. A facility for tracing data traffic on a network, the facility comprising: an identifying means for identifying a process causing a transmission or receipt of a communication via the network; and a logging means in communication with the identifying means for logging and event, wherein the event comprises the identification the process and wherein the logging means is useable to determine the volume of data traveling over the network.

11. The apparatus of claim 10 wherein the identifying means further comprises means for communicating with a transport layer of a protocol stack.

12. A computer-readable medium having stored thereon computer-executable instructions for performing steps comprising: at the transport layer of a protocol stack residing on a first device in the network, detecting a transmission or receipt of data to or from a second device on the network; and in response to the transmission or receipt being

detected, recording the transmission or receipt as an entry in a trace log, wherein the trace log is accessible to determine the volume of data traveling over a network.

13. The computer-readable medium of claim 12, wherein the protocol stack is a TCP/IP stack.

14. The computer-readable medium of claim 12, having further computer-executable instructions for performing the step of detecting the presence of an input/output packet representing the transmission or receipt.

Sub B6

15. A computer-readable medium having stored thereon computer-executable instructions for performing steps comprising: detecting the presence of an input/output packet requesting a transmission; searching the input/output request packet to determine the identity of the process that created the input/output request packet; and storing in a trace log an entry representing the transmission, wherein the entry comprises the identity of the process, and wherein the trace log is accessible to determine the volume of data being transmitted over the network.

16. The computer-readable medium of claim 15, having further computer-executable instructions for performing the step of detecting the presence of the input/output packet at the transport layer of a protocol stack.

17. The computer-readable medium of claim 15, having further computer-executable instructions for performing the step of detecting an acknowledgment of the transmission; and in response to the detection of the acknowledgment, storing in the trace log an entry representing the completion of the transmission.

Sub B7

18. A computer-readable medium having stored thereon computer-executable instructions for performing the steps comprising: detecting the presence of a packet for an input-output connection to a port; searching the packet to determine the identity of the process that created the packet; and in response to the detection of a receipt of data at the port, storing in a trace log an entry representing the receipt of the data, wherein the entry

Cont Sub B7

comprises the process identification, and wherein the trace log is accessible to determine the volume of the data being transmitted over the network.

19. The computer-readable medium of claim 18, having further computer-executable instructions for performing the steps of: creating a connection object representing the opening of the port connection by the process; copying the process identification from the connection object into a transport control block associated with the port; and in response to the detection of the receipt of data at the port, copying the process identification into the trace log.

20. The computer-readable medium of claim 18, having further computer-executable instructions for performing the step of copying the process identification from the connection object into the transport control block so that the process identification is contiguous with the rest of the data in the transport control block.

21. The computer-readable medium of claim 18, having further computer-executable instructions for performing the steps of: detecting the presence of an input/output request packet indicating that the data receipt is complete; and in response to the detection of the completion input/output request packet, storing in the trace log an entry representing the receipt of the data.

Add B8

Appendix A:

Event Name	TID	Clock (ms)	Kt	Ut	Parent TID	Parent PID	Sz	Image Name		
ProcessStart	1C0	1124570445	2	0	10C	34C		Tracelog.exe		
ThreadStart	1C0	1124570445	2	0	1C0	10C				
ThreadEnd	1C0	1124570492	3	0	1C0	10C				
ProcessEnd	1C0	1124570492	3	0	10C	34C		Tracelog.exe		
DiskWriteIo	14	1124571070	0	0	2	67	300			
ProcessStart	2AC	1124572773	36	28	1C0	288		CMD.exe		
ThreadStart	2AC	1124572773	36	28	10C	1C0				
ThreadStart	10C	1124572898	0	2	364	1C0				
Event Name	**TID**	**Clock (ms)**	**Kt**	**Ut**	**Source Addr**	**Dest Addr**	**Sport**	DPort	**Size**	**PID**
TcpSend	0	1124572976	26425	0	172.31.249.34	172.31.255.147	5010	5020	8192	1C0
TcpSend	0	1124572976	26425	0	172.31.249.34	172.31.255.147	5010	5020	8192	1C0
TcpSend	0	1124572976	26425	0	172.31.249.34	172.31.255.147	5010	5020	8192	1C0
TcpSend	0	1124572992	26426	0	172.31.249.34	172.31.255.147	5010	5020	8192	1C0
TcpSend	0	1124572992	26426	0	172.31.249.34	172.31.255.147	5010	5020	8192	1C0
TcpSend	0	1124573008	26426	0	172.31.249.34	172.31.255.147	5010	5020	8192	1C0
TcpSend	0	1124573008	26426	0	172.31.249.34	172.31.255.147	5010	5020	8192	1C0
TcpSend	0	1124573023	26427	0	172.31.249.34	172.31.255.147	5010	5020	8192	1C0
TcpRecv	***0***	***1124573093***	***78293***	***1***	***172.31.249.34***	***172.31.254.11***	***80***	***6437***	***531***	***382***
TcpRecv	***0***	***1124573164***	***78294***	***3***	***172.31.249.34***	***172.31.254.11***	***80***	***6437***	***164***	***382***
TcpSend	***39C***	***1124573198***	***78296***	***0***	***172.31.249.34***	***172.31.254.11***	***80***	***6437***	***1507***	***382***
TcpSend	***39C***	***1124573231***	***78303***	***0***	***172.31.249.34***	***172.31.254.11***	***80***	***6437***	***9236***	***382***
TcpSend	***39C***	***1124573362***	***78304***	***0***	***172.31.249.34***	***172.31.254.11***	***80***	***6437***	***6728***	***382***
TcpSend	0	1124573570	26460	0	172.31.249.34	172.31.255.147	5010	5020	8192	1C0
TcpSend	0	1124573586	26461	0	172.31.249.34	172.31.255.147	5010	5020	8192	1C0
TcpSend	0	1124573586	26461	0	172.31.249.34	172.31.255.147	5010	5020	8192	1C0
TcpSend	0	1124573601	26462	0	172.31.249.34	172.31.255.147	5010	5020	8192	1C0
TcpSend	0	1124573601	26462	0	172.31.249.34	172.31.255.147	5010	5020	8192	1C0
TcpSend	0	1124573617	26463	0	172.31.249.34	172.31.255.147	5010	5020	8192	1C0
TcpSend	0	1124573617	26463	0	172.31.249.34	172.31.255.147	5010	5020	8192	1C0
TcpSend	0	1124573633	26464	0	172.31.249.34	172.31.255.147	5010	5020	8192	1C0
TcpSend	0	1124573648	26465	0	172.31.249.34	172.31.255.147	5010	5020	8192	1C0
TcpSend	0	1124573648	26465	0	172.31.249.34	172.31.255.147	5010	5020	8192	1C0

Event Name	TID	Clock (ms)	Kt	Ut	Parent TID	Parent PID	Sz	Image Name
ThreadEnd	364	1124573758	0	0	364	1C0		
ThreadEnd	10C	1124573789	1	3	10C	1C0		
ProcessEnd	10C	1124573789	1	3	1C0	288		nttcp.exe

Legend: Kt - Kernel Mode Time

Ut - User Mode CPU Time

PID - Process ID

TID - Thread ID

The following are the extended record fields for the respective events:

- Process start - new Process Id, its Parent's Process id
- Process end - current Process Id, its Parent's Process id, the image filename that it was running
- Thread start - new thread Id, its Process Id
- Thread end - current thread Id being terminated, its Process Id.
- I/O read, I/O write - The signature of the disk where the I/O operation was done, the transfer size.
- TCPSend - Source Address, Destination Address, Source port, Destination port, Size, ProcessId